

Client Credential Flow Documentation

Mavim API

Access Mavim API as a Daemon Application

Start

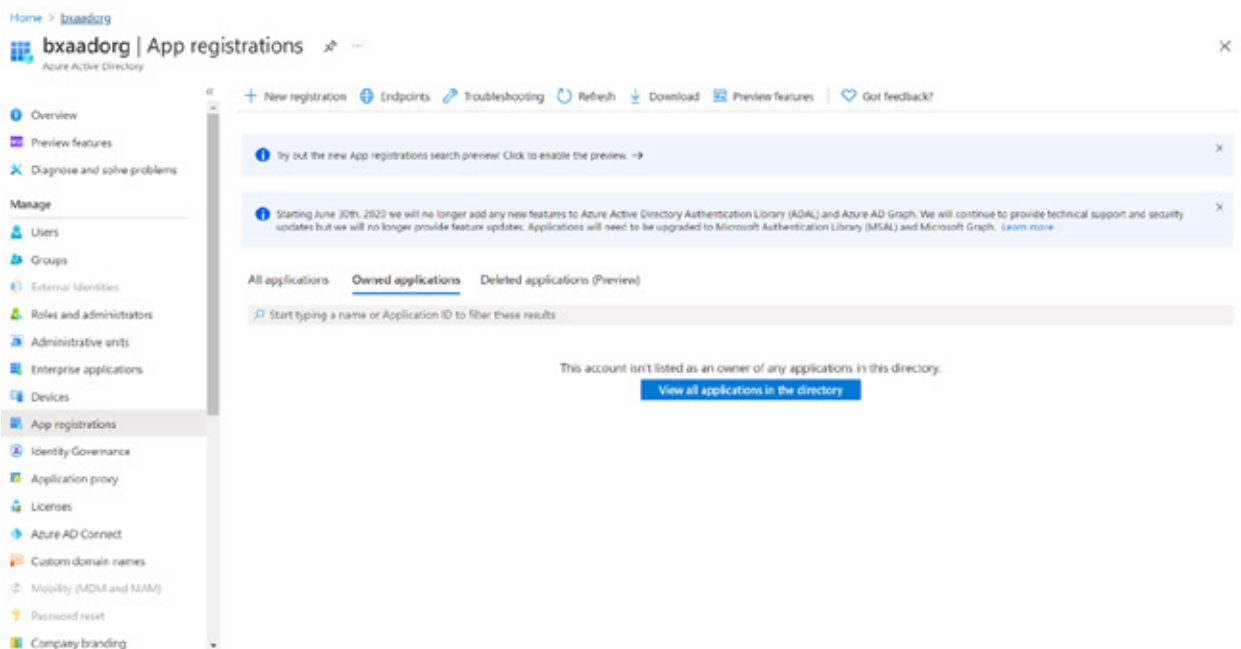
Creating app registration

Sign in to Azure portal: Open your web browser and navigate to Azure portal (<https://portal.azure.com>). Enter your Azure account credentials (username and password) and sign in.

Navigate to Azure active directory: After signing in, you'll land on the Azure dashboard. On the left-hand sidebar, locate and click on the 'Azure Active Directory' option.

Select 'App registrations': In the Azure active directory pane, under the 'Manage' section, click on 'App registrations'.

Create a new app registration: Once in the 'App registrations' page, you'll find a button labeled 'New registration' at the top. Click on it.



Configure the registration: In the 'Register an application' page; Enter a meaningful name for your application in the 'Name' field. This name helps you identify your application within Azure.

Register the application: After configuring the necessary details, click on the 'Register' button at the bottom of the page. This action registers your application with Azure AD.

Access Mavim API as a Daemon Application

Home > b2aad020 > Register an application

Name
The user-facing display name for this application (this can be changed later).
myhealthapiapp

Supported account types
Who can use this application or access this API?
☒ Accounts in this organizational directory only (b2aad020- Single tenant)
☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
☐ Personal Microsoft accounts only
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
Web e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

Note down the application (client) ID and Object ID: Once the registration is successful, you'll be redirected to the overview page of your newly created app registration. On this page, you'll find the 'Application (client) ID'. This is a unique identifier for your application within Azure AD. Copy this ID and store it securely as you'll need it later. Also copy the 'Object ID' for later use.

Access Mavim API as a Daemon Application

The top screenshot shows the Microsoft Identity Platform (Azure AD) interface for the application 'myhealthapisapp1'. The left sidebar contains navigation options: Overview, Quickstart, Integration assistant, Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators [Preview], Manifest), and Support + Troubleshooting (Troubleshooting, New support request). The main content area displays the 'Essentials' section with the following information:

- Display name: myhealthapisapp1
- Application (client) ID: [redacted]
- Object ID: [redacted]
- Directory (tenant) ID: [redacted]
- Supported account types: My organization only
- Client credentials: Add a certificate or secret
- Redirect URIs: Add a Redirect URI
- Application ID URI: Add an Application ID URI
- Managed application in L: myhealthapisapp1

The bottom screenshot shows the 'myhealthapisapp1' app registration overview page. The left sidebar contains navigation options: Overview, Deployment Plan, Manage (Properties, Owners, Roles and administrators [Preview], Users and groups, Single sign-on, Provisioning, Application proxy, Self service), Security (Conditional Access, Permissions, Token encryption), Activity (Sign-ins, Usage & insights, Audit logs), and What's New. The main content area displays the 'Properties' section with the following information:

- Name: myhealthapisapp1
- Application ID: 5969ade1-0b19-4d56-a122-...
- Object ID: acf2215f-c498-453e-b677-1...

The 'Getting Started' section includes three steps:

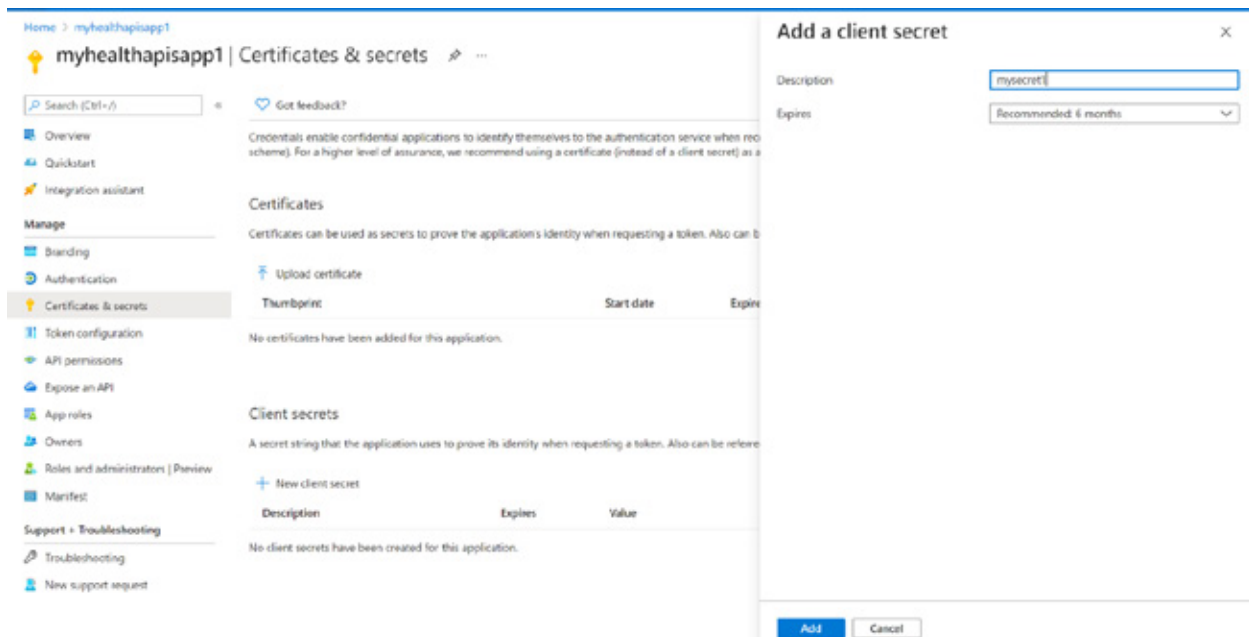
- 1. Assign users and groups**: Provide specific users and groups access to the application. [Assign users and groups](#)
- 2. Provision user accounts**: You'll need to create user accounts in the application. [Learn more](#)
- 3. Self service**: Enable users to request access to the application using their Azure AD credentials. [Get started](#)

The 'What's New' section includes two items:

- Sign in charts have moved!**: The new insights view shows sign in info along with other useful application data. [View insights](#)
- Delete Application has moved to Properties**: You can now delete your application from the Properties page. [View properties](#)

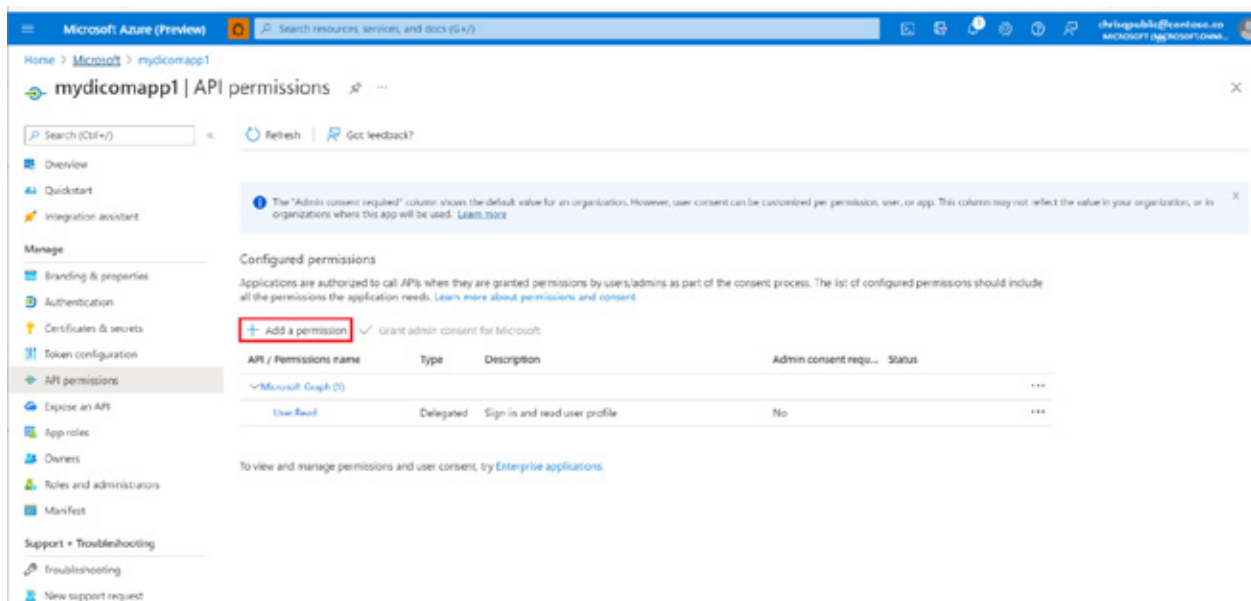
Generate and note down application secret: In the left-hand sidebar of your app registration, under 'Manage', click on 'Certificates & secrets'. Here, you can add new client secret by clicking on '+ New client secret'. Enter a description for the secret, choose the maximum expiration duration, and click on the 'Add' button. After the secret is generated, note down the secret value immediately. This secret is required for authenticating your application.

Access Mavim API as a Daemon Application



Set up API permissions: Under the 'Manage' in the left-hand sidebar, select 'API permissions'. Click on '+ Add a permission' to grant your app access to APIs that it needs to interact with. Follow the prompts to select the APIs and the specific permissions required for your application's functionality. Select Mavim.iMprove.App.ReadWrite.All permissions here.

Access Mavim API as a Daemon Application



Request API permissions

Select an API

Microsoft APIs APIs my organization uses My APIs

Apps in your directory that expose APIs are shown below

Azure API for DICOM	
Name	Application (client) ID
Azure API for DICOM	75e725bf-66ce-4cea-9b9a-5c4cae57f33

Grant admin consent: If your application requires admin consent for the permissions, you'll see a notification on the API permissions page. Click on the 'Grant admin consent' button to grant consent. This allows your application to access resources on behalf of users.

Access Mavim API as a Daemon Application

Home > Enterprise applications | All applications > Microsoft Entra SAML Toolkit 1

Microsoft Entra SAML Toolkit 1 | Permissions

Enterprise Application

« Refresh Review permissions Got feedback?

Manage

- Properties
- Owners
- Roles and administrators
- Users and groups
- Single sign-on
- Provisioning
- Self-service
- Custom security attributes

Security

- Conditional Access
- Permissions
- Token encryption

Activity

- Sign-in logs
- Usage & insights

Permissions

Below is the list of permissions that have been granted for your organization. As an administrator, you can grant permissions to this app on behalf of all users (delegated permissions). You can also grant permissions directly to this app (app permissions). [Learn more.](#)

You can review, revoke, and restore permissions. [Learn more.](#)

To configure requested permissions for apps you own, use the [app registration](#).

Grant admin consent for Contoso

Admin consent

Search permissions

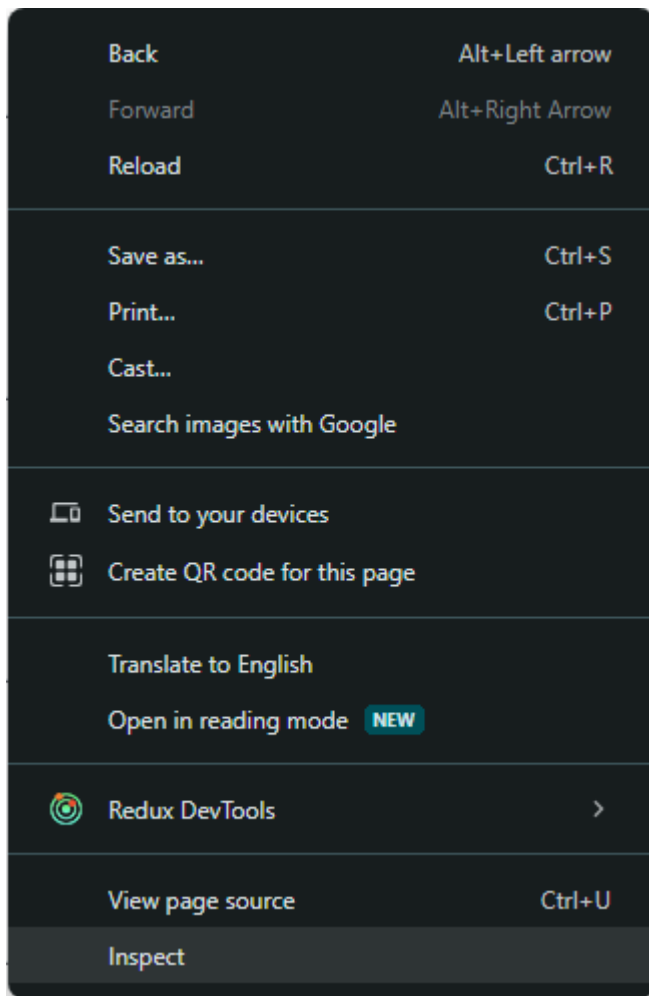
API Name	↑↓	Claim value	↑↓	Permission	↑↓	Type	↑↓	Granted
Microsoft Graph								
Microsoft Graph		offline_access		Maintain access to dat...		Delegated		Admin c...
Microsoft Graph		openid		Sign users in		Delegated		Admin c...
Microsoft Graph		Application.Read.All		Read all applications		Application		Admin c...

Once you've completed these steps, your app registration is set up. Make sure to securely store any sensitive information such as client secret to ensure the security of your application.

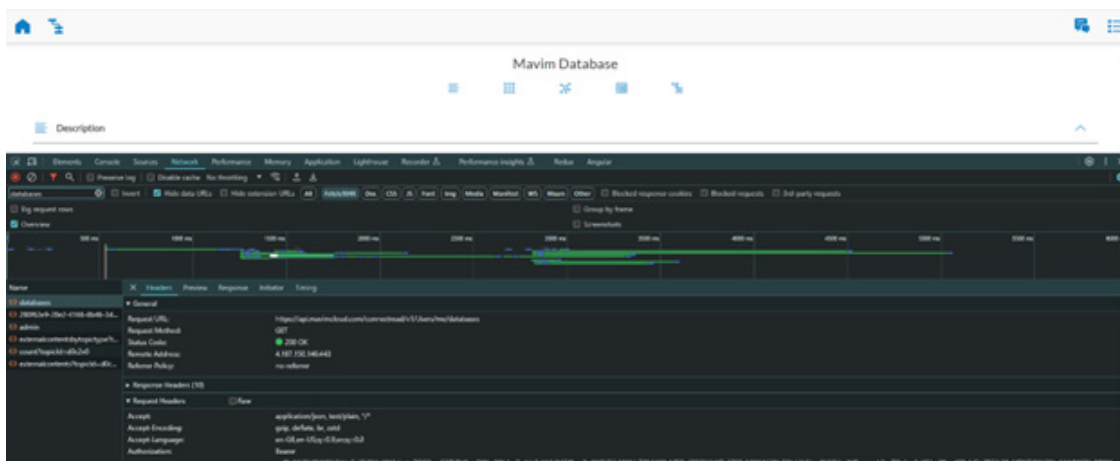
Get global admin token (User only access token)

Access <https://improve.mavimcloud.com/> as the global admin user. Right click from application and select *Inspect*;

Access Mavim API as a Daemon Application



Developer tools window will be opened. Go to *Network* tab. Reload the page. Click on *Fetch/XHR* button. Enable filter and type 'databases'. Select *databases* where the *Type* reads *xhr*. From the *Headers* tab, under the *Request Headers*, copy the value in *Authorization* excluding 'Bearer'. This is your global admin token.



Access Mavim API as a Daemon Application

Client credentials flow/Get application token (App only access token)

Get token using Postman: Trigger a GET request using Postman with the below credentials.

GET - <https://login.microsoftonline.com/<your tenant ID goes here>/oauth2/token>

Body -> x-www-form-urlencoded

grant_type: client_credentials

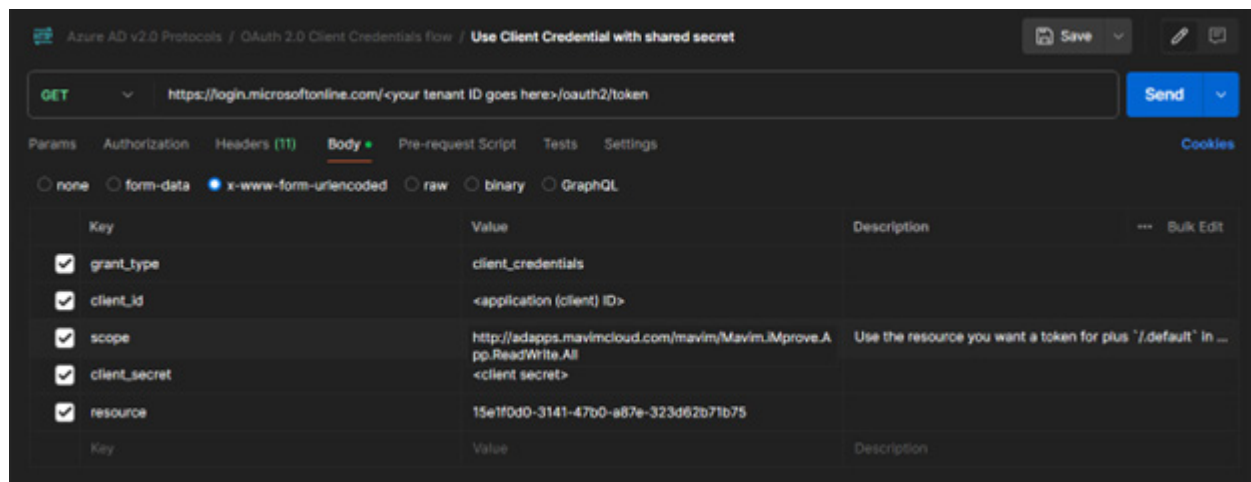
client_id: <application (client) ID noted down in earlier step goes here>

scope: <http://adapps.mavimcloud.com/mavim/Mavim.iMprove.App.ReadWrite.All>

client_secret: <client secret noted down in earlier step goes here>

resource: 15e1f0d0-3141-47b0-a87e-323d62b71b75 (Mavim iMprove app's application (client) ID)

Send the GET request and copy the token from response.



Get the sub from token: Paste the token received from previous step in Encoded section of <https://jwt.io/> and note down the value of 'sub' property from the Decoded section.

Access Mavim API as a Daemon Application

Algorithm

HS256

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "iat": 1516239022}
```

VERIFY SIGNATURE

```
HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload), secret)
```

Access Mavim API as a Daemon Application

Onboard application

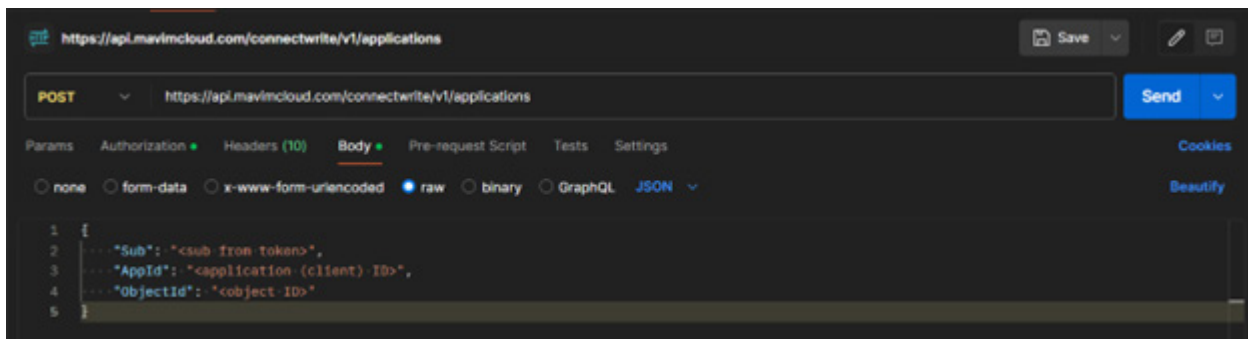
Create application endpoint: POST <https://api.mavimcloud.com/connectwrite/v1/applications>

Body -> raw -> JSON

```
{
  "Sub": "<sub noted down from previous step goes here>",
  "AppId": "<application (client) ID noted down from earlier step goes here>",
  "ObjectId": "<object ID noted down from earlier step goes here>"
}
```

Use global admin token for authorization.

This would return the application ID (GUID) inserted in database, note down the GUID for later use.

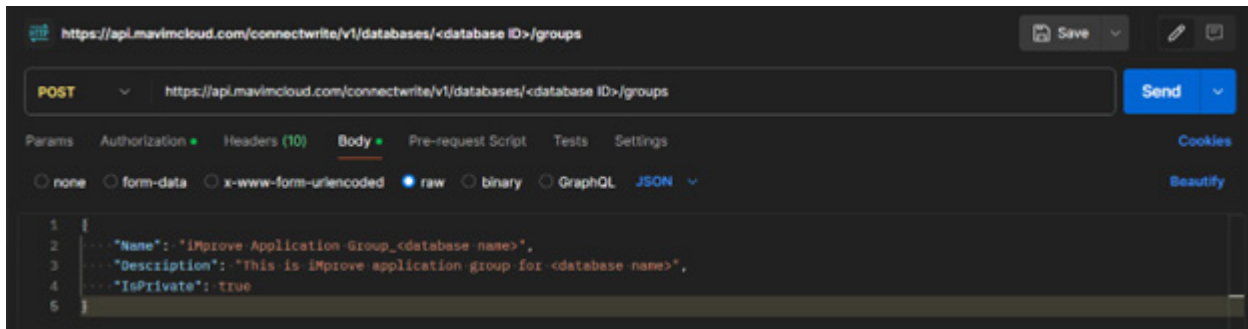


Create application group endpoint: POST <https://api.mavimcloud.com/connectwrite/v1/databases/<your database ID goes here>/groups>

Body -> raw -> JSON

```
{
  "Name": "iMprove Application Group_<your database name goes here>",
  "Description": "This is iMprove application group for <your database name goes here>",
  "IsPrivate": true
}
```

Access Mavim API as a Daemon Application



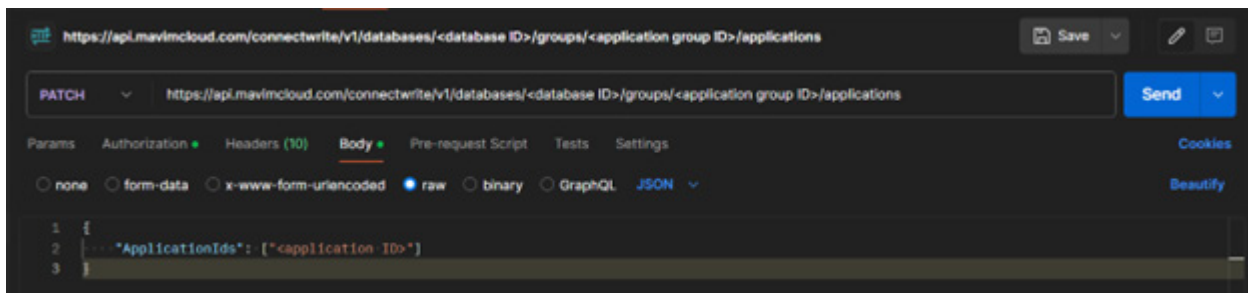
Use global admin token for authorization.

This would return the group ID (GUID) inserted in database, note down the GUID for later use.

Link application to application group endpoint: PATCH `https://api.mavimcloud.com/connectwrite/v1/databases/<your database ID goes here>/groups/<application group ID returned from previous step goes here>/applications`

Body -> raw -> JSON

```
{
  "ApplicationIds": ["<application ID returned from earlier step>"]
}
```



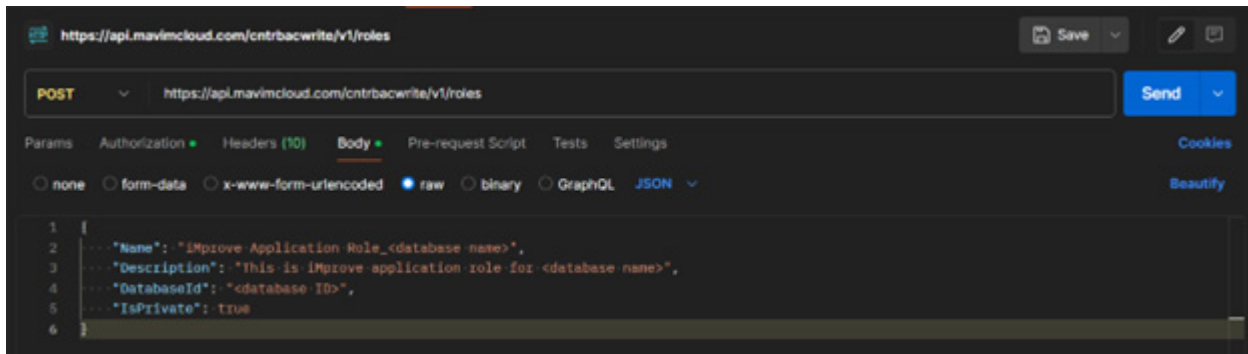
Use global admin token for authorization.

Create application role in Connect RBAC endpoint: POST `https://api.mavimcloud.com/cntrbacwrite/v1/roles`

Body -> raw -> JSON

```
{
  "Name": "iMprove Application Role_<your database name goes here>",
  "Description": "This is iMprove application role for <your database name goes here>",
  "DatabaseId": "<your database ID goes here>",
  "IsPrivate": true
}
```

Access Mavim API as a Daemon Application



Use global admin token for authorization.

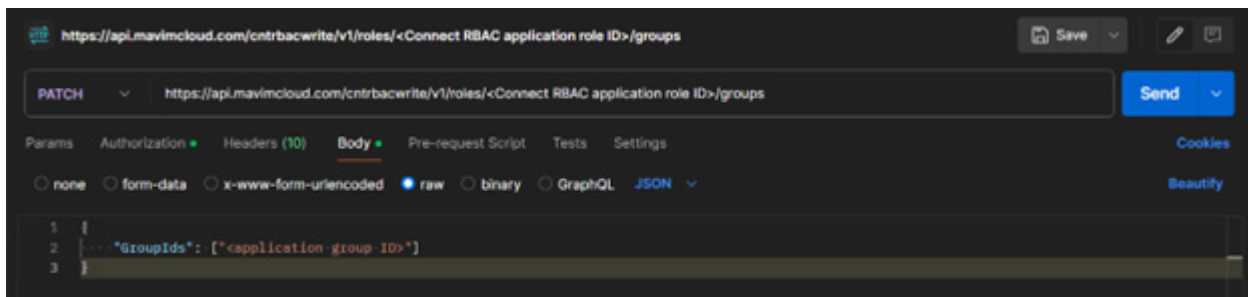
This would return the role ID (GUID) inserted in database, note down the GUID for later use.

Link application group to Connect RBAC application role endpoint: PATCH <https://api.mavimcloud.com/cntrbacwrite/v1/roles/<application role ID returned from previous step goes here>/groups>

Body -> raw -> JSON

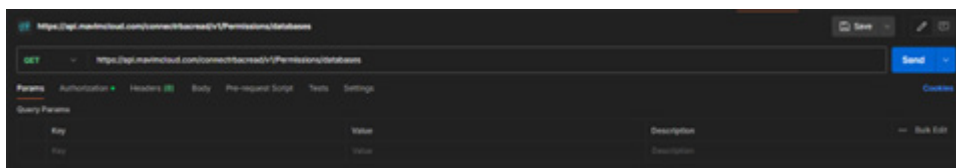
```
{
  "GroupIds": ["<application group ID noted down from earlier step goes here>"]
}
```

Use global admin token for authorization.



Get database permission ID endpoint: GET

<https://api.mavimcloud.com/connectrbacread/v1/Permissions/databases>



Use global admin token for authorization.

Access Mavim API as a Daemon Application

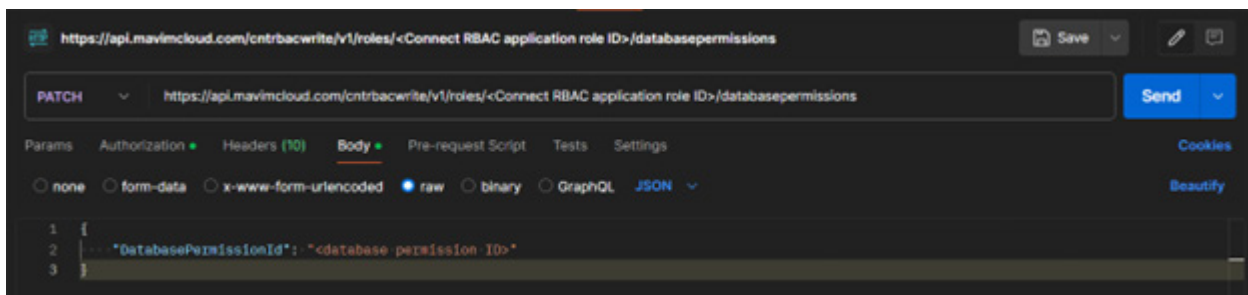
This would return the database permissions exist under your company. Find the JSON object with the database name you want to give permission to the application for, and copy the GUID value from id property.

Link database permission to Connect RBAC application role endpoint: PATCH <https://api.mavimcloud.com/cntrbacwrite/v1/roles/<Connect RBAC application role ID returned from previous step goes here>/databasepermissions>

Body -> raw -> JSON

```
{
  "DatabasePermissionId": "<database permission ID copied from previous step goes here>"
}
```

Use global admin token for authorization.



Create iMprove RBAC application role endpoint: POST <https://api.mavimcloud.com/authwrite/v1/databases/<your database ID goes here>/roles>

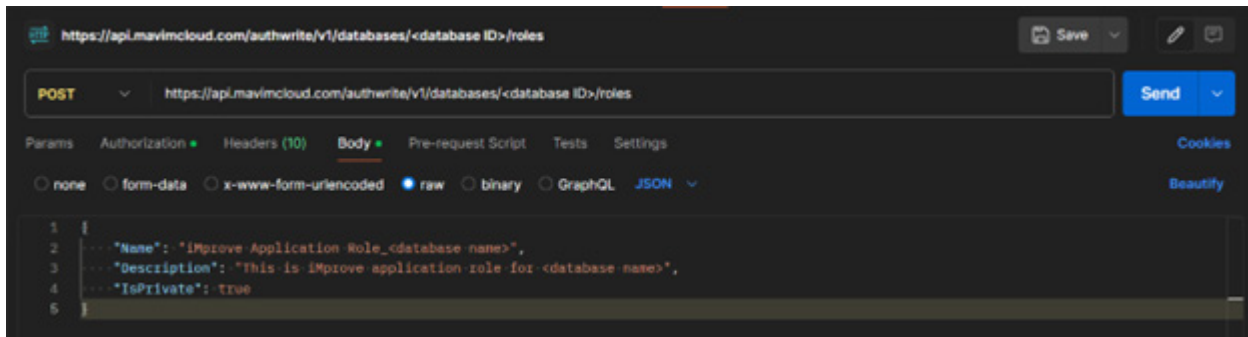
Body -> raw -> JSON

```
{
  "Name": "iMprove Application Role_<your database name goes here>",
  "Description": "This is iMprove application role for <your database name goes here>",
  "IsPrivate": true
}
```

Use global admin token for authorization.

This would return the role ID (GUID) inserted in database, note down the GUID for later use.

Access Mavim API as a Daemon Application

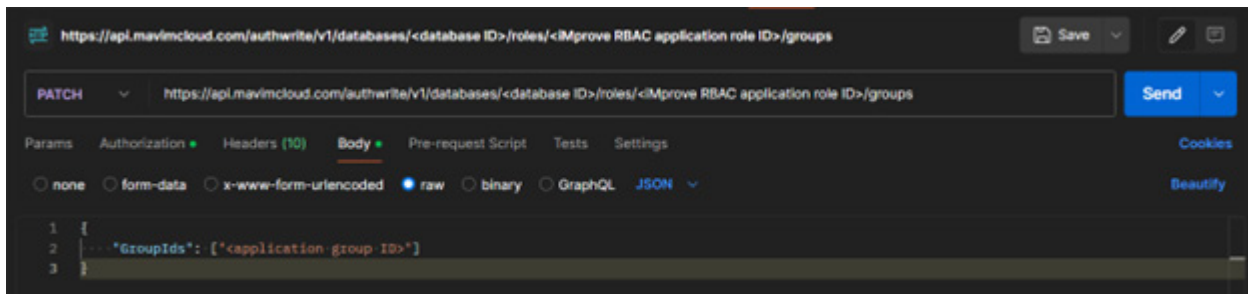


Link application group to iMprove RBAC application role endpoint: PATCH `https://api.mavimcloud.com/authwrite/v1/databases/<your database ID goes here>/roles/<application role ID returned from previous step goes here>/groups`

Body -> raw -> JSON

```
{
  "GroupIds": [ "<application group ID returned from earlier step goes here>" ]
}
```

Use global admin token for authorization.



Create topic permission endpoint: POST

`https://api.mavimcloud.com/authwrite/v1/databases/<your database ID goes here>/permissions/topic`

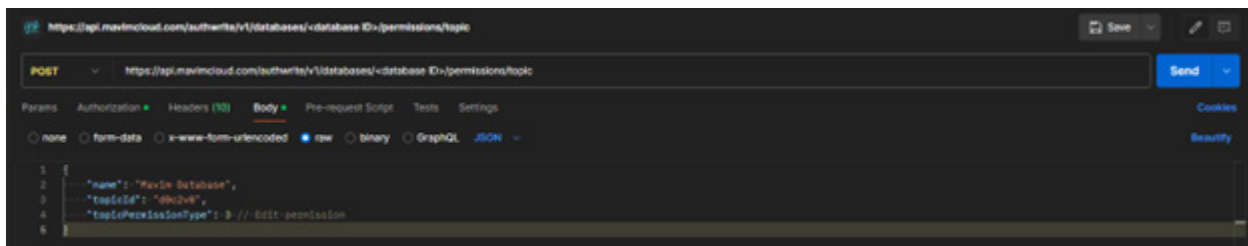
Body -> raw -> JSON

```
{
  "name": "Mavim Database",
  "topicId": "d0c2v0",
  "topicPermissionType": 3 // Edit permission
}
```

Access Mavim API as a Daemon Application

Note:

- 0 - Topic will not be visible to the application
- 1 - Topic title will be visible to the application
- 2 - Topic data will be visible to the application
- 3 - Application can modify topic data



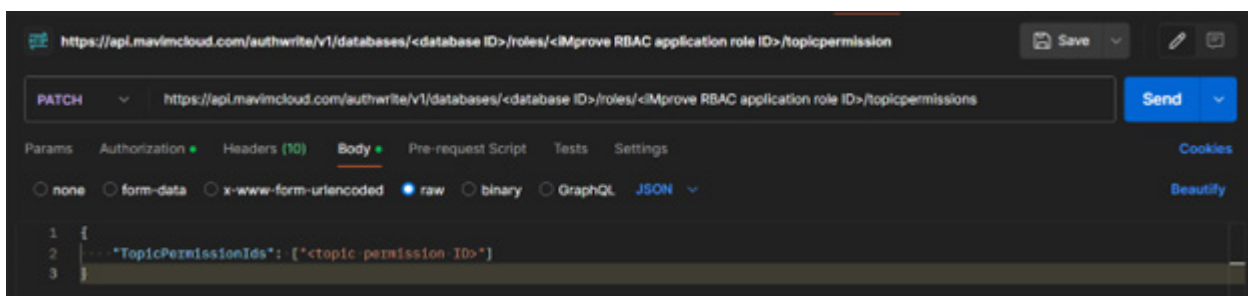
Use global admin token for authorization.

This would return the topic permission ID (GUID) inserted in database, note down the GUID for later use.

Link topic permission to iMprove RBAC application role endpoint: PATCH `https://api.mavimcloud.com/authwrite/v1/databases/<your database ID goes here>/roles/<iMprove RBAC application role ID returned from earlier step goes here>/topicpermissions`

```
{ 1: "TopicPermissionIds": ["<topic permission ID returned from previous step goes here>"] 2: }
```

Use global admin token for authorization.



Access Mavim API as a Daemon Application

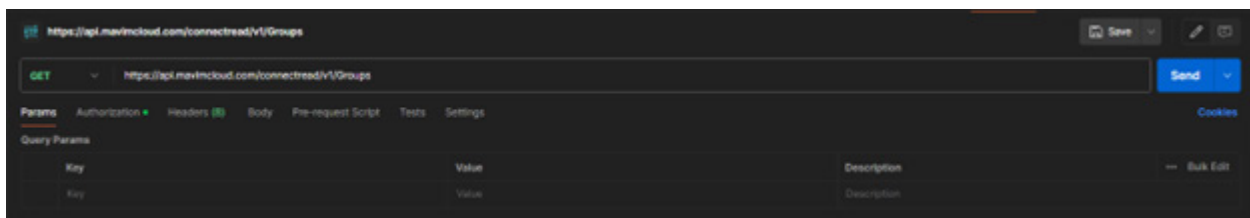
Assign admin permission to application (alternate flow)

Create application endpoint: Follow the same steps from Onboard application section

Get admin group ID endpoint: GET

<https://api.mavimcloud.com/connectread/v1/Groups>

Use global admin token for authorization.



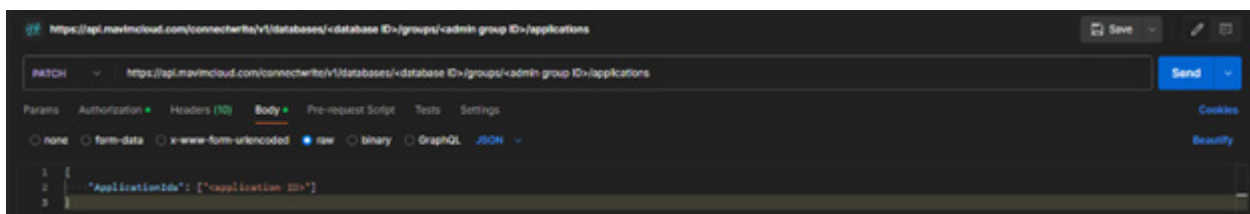
This would return the groups exist under your company. Find the JSON object with the group name `improveAdmins__<your database name>` you want to link application to, and copy the GUID value from `id` property.

Link application to admin group endpoint:

<https://api.mavimcloud.com/connectwrite/v1/databases/<your database ID goes here>/groups/<admin group ID copied from previous step goes here>/applications>

Body -> raw -> JSON

```
{
  "ApplicationIds": ["<application ID returned from first step>"]
}
```



Use global admin token for authorization.

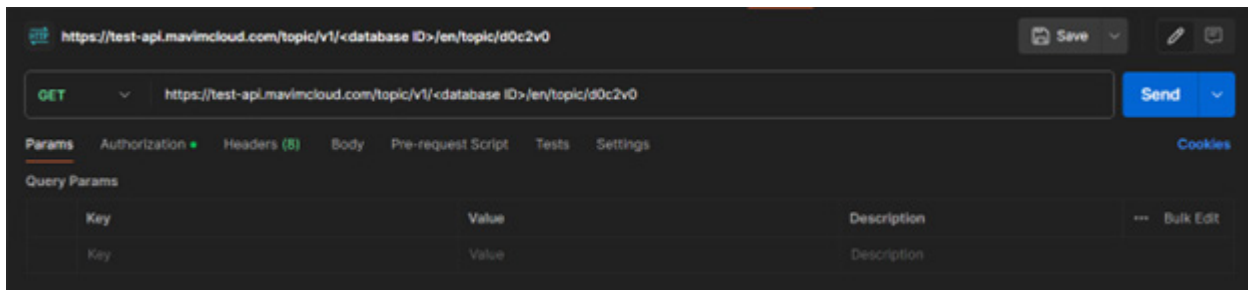
Access Mavim API as a Daemon Application

Fetch topic information

Get topic endpoint: GET <https://api.mavimcloud.com/topic/v1/<your database ID goes here>/en/topic/d0c2v0>

Use application token for authorization.

Topic data in response proves your accessibility to Mavim API on behalf of user as an application.



End

GET & STAY CONNECTED

youtube.com/mavimtv

2.75 K

You
Tube

linkedin.com/company/mavim

3.45 K

in

twitter.com/MavimBV

1.6 K

X

mav!m
change it

www.mavim.com